RP 507 Wind Turbine Condition Based Maintenance System Open Architecture

The following recommended practice (RP) is subject to the disclaimer at the front of this manual. It is important that users read the disclaimer before considering adoption of any portion of this recommended practice.

This recommended practice was prepared by a committee of the AWEA Operations and Maintenance (O&M) Committee.

Committee Chair: David Zeglinski, OSIsoft, LLC
Principal Author: Kevin Line, Sentient Science

Purpose and Scope

The purpose of this document is to provide a recommended practice for condition based maintenance (CBM) system architecture for the wind plant, including wind turbine generator, balance of plant, and other elements.

Condition based maintenance and condition monitoring has been shown in many industries to reduce the cost of ownership and increase the availability of assets for operations. Aviation and energy have multiple examples of implementing CBM with positive financial and operational results.

The goal of the best practice is to provide a common, scalable, and open architecture to enable an interoperability and cooperation for CBM systems. The advantage to this approach is that the wind plant operator and owner will be able to leverage best-in-breed approaches for CBM through the implementation of an open architecture approach. Furthermore, future technology and capability will be easily integrated into the system with little need for reconfiguration or modification.

Introduction

Implementation of a condition monitoring system can take many forms, processes, and approaches. The general diagram of these systems is shown in Figure A. The description of each component is as follows:

1. Wind Plant: The collection of wind turbine generators and balance of plant equipment needed to generate electricity.

2. Wind Turbine Generator (WTG): The electrical and mechanical system for converting wind energy into electrical energy, including tower, foundation, and balance of plant. The control center for the WTG is not included.

Introduction
(continued)

3. Balance of plant (BOP): Remaining hardware in the plant, not including the WTG.

4. Control Sensors & Hardware: The hardware and software system, typically the SCADA system, on the WTG which supports the control and operation.

5. Condition Monitoring & Sensors: Data collection, processing, and sensors for the purposes of assessing the health and remaining life. This equipment is in addition to SCADA hardware.

6. Data Storage: Standards-based storage of health and control data for the purpose of condition monitoring. Stored data can be local, centralized, or both, depending on system architecture.

7. Data Processing: Local or remote health management system processes collected real-time or off-line data either in time or frequency domain. Health metrics and indicators are restored in the database.

8. Maintenance Interface: Alerts, health indicators, and actions are communicated to appropriate stakeholders, ranging from local maintenance management to supply chain and engineering.

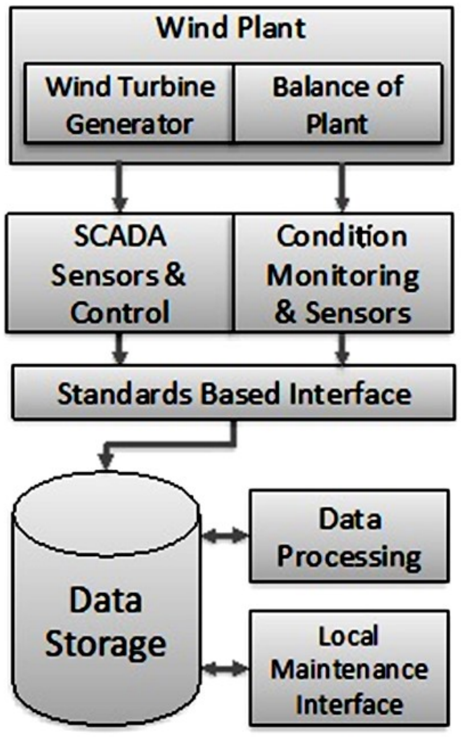

Figure A: Basic CBM System Architecture

CBM Open Architecture

1. CBM System Development

CBM system and data management strategy will be unique for each platform and plant, but follow a similar process. This process is outlined below:

1.1. Identify system failure modes for both WTG and BOP through review of system supplier failure modes and effects (FMEA) analysis, industry data, and interviews with subject matter experts.

1.2. Determine CBM needs and strategy through analysis. Identify high priority components for CBM with careful consideration of failure rate, replacement cost, spare part lead-time, and impact on operations.

1.3. For wind plant, determine required sensors, data collection, processing, and storage equipment to meet strategy.

1.4. Leverage open architecture for CBM system and data management. Through application of open architecture, data collection, management, and processing will have common interfaces to each development and integration. This architecture is created through the open standards approach outlined in the sections below.

1.5. Implement CBM system, through procurement of hardware and software. Install systems and configure per manufacturer instructions.

2. Open Standards: MIMOSA

Once CBM needs have been defined for the system, open standards should be applied. MIMOSA publishes a well-accepted open standard for developing and implementing condition based maintenance systems. Both the OSA-CBM and OSA-EAI are data and communication architectures that define the interfaces between hardware and software. These common interface definitions enable the application of 3rd party capabilities built to the same interface definition and enable data, software, and hardware to remain compatible well into the future, as along as the standards are applied. The complete definitions are found at *www.mimosa.org*. These open standards are the basis of this recommended practice.

The OSA-CBM architecture is defined by a set of components (physical or virtual components in the system) and workflows (transportation of data from source to user). To achieve this, the architecture is composed of segments and agents. Segments correspond to measurement locations (sensors), and agents (people or systems that analyze data).

2. Open Standards: MIMOSA
(continued)

The workflow for this system is conceptualized in Figure B. From the point of view of the CBM framework, each sensor would be a measurement location. To populate the ports in the module with data acquisition (DA) data events, the sensor interface would be wrapped in an algorithm. These ports and any DA data events they contain would then be available for the rest of the Configuration to use. By using the ports of one or more algorithms as input for other algorithms, the configuration specifies a workflow that processes the data as it flows through. Additional pre-processing for the measurement locations is done at the data manipulation and state determination levels, producing corresponding data events. The end products of this workflow are health assessment, prognostics assessment and advisory generation data events. These high-level data events are created by agents, interfaced by algorithms in the workflow that provide interpretations of the health and prognosis, and provide recommendations on how to deal with them. The CBM process makes data events available to external processes via the interface types in the OSA-CBM specification.

For example, onboard the WTG, sensor data would be stored as data acquisition events. During operation, the bandwidth usage would be minimized by limiting the data events sent to the ground station with monitor ID groups to pass on DA events, filtering out those with number alerts below a certain severity. Additional data may be requested by passing a monitor ID group to a CBM interface requesting a specific subset of data. They would be transferred in a serialized (XML, JSON, YAML, etc.) compressed format.

Another possibility is to move some of the more critical or less processor intensive algorithms in the workflow onboard the WTG. Health and prognostics assessment data events produced by an onboard digital twin would require much less bandwidth than the data acquisition events consumed to produce them. This flexibility allows better balancing of the tradeoff between onboard processing and platform to ground station bandwidth. The network of algorithms and ports produce and consume the data manipulation, state detection, health assessment, prognostics assessment, and advisory generation data events. External applications may then request the data events from specific ports provided by the CBM interface by using monitor ID groups. HA events are used to provide the health level of components; PA events report their remaining useful lives; and AG events give recommendations and optionally requests for work.

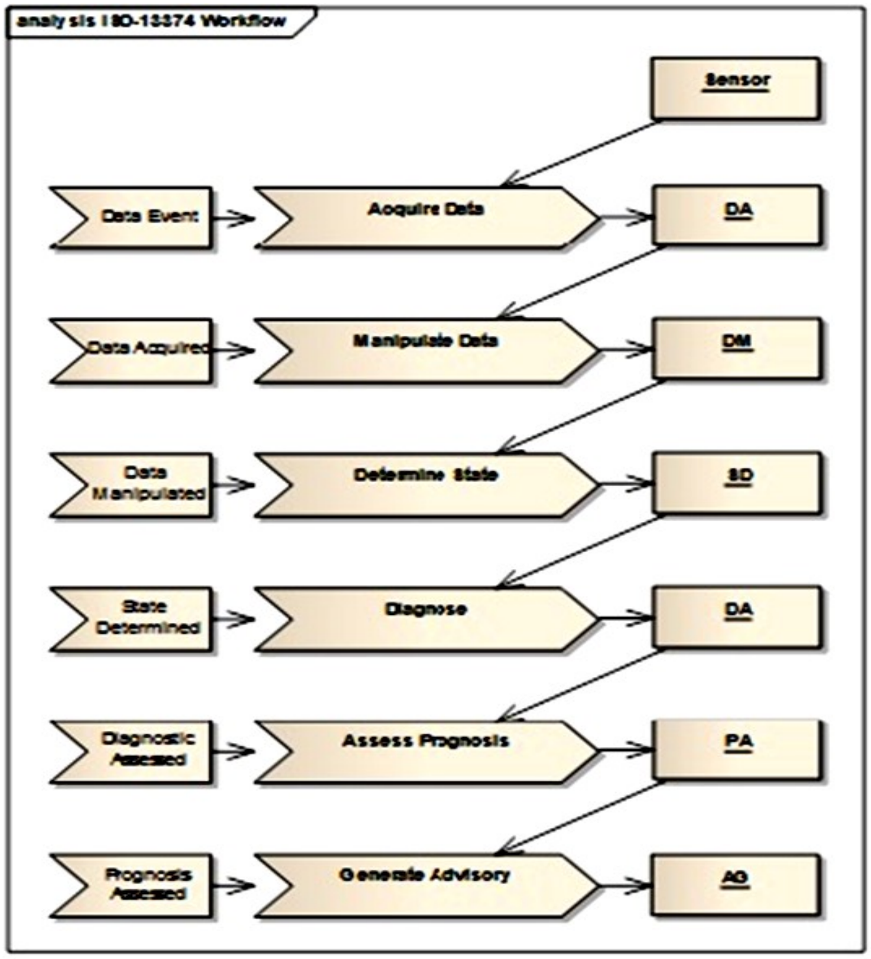

Figure B: OSA-CBM Workflow

3. Interface Definitions

Condition based maintenance system elements will be implemented in the above architecture using the OSA-CBM interface definitions. OSA-CBM interface definitions simplify integrating a wide variety of software and hardware components, as well as developing a framework for these components by specifying a standard architecture and framework for implementing condition based maintenance systems. They describe the functional blocks of CBM systems, as well as the interfaces between those blocks. The standard provides a means to integrate many disparate components, including interfaces with sensors, data acquisition devices, and software algorithms and eases the process by specifying the inputs and outputs between the components. In short, it describes a standardized information delivery system for condition based monitoring. It describes the information that is moved around and how to move it. It also has built in meta-data to describe the processing that is occurring.

3. Interface Definitions
(continued)

OSA-CBM provides an interface standard and defines the interfaces between the functional blocks in a CBM system. Vendors can develop algorithms to fit inside of these blocks, separating the information processing from how it is presented. This separation allows proprietary code and algorithms to be kept hidden inside each of the functional blocks. It also creates a plug-and-play capability where vendors can easily insert updates or roll back to previous versions without affecting other modules or programs relying on the functional blocks. Figure C illustrates an example of proprietary algorithm in one OSA-CBM block.

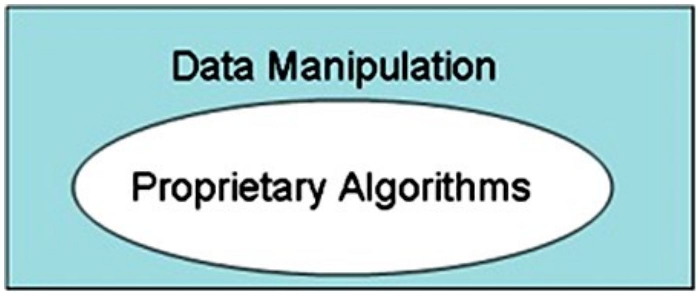

Figure C: Example of Proprietary Algorithm in One OSA-CBM Block

Figure D illustrates the main part of algorithm configuration in OSA-CBM UML specification 3.3.0. Configuration provides information about algorithm input data, descriptions of algorithms used for processing input data, a list of outputs, and various output specifics, such as engineering units and thresholds for alerts.

Writers of algorithms simply need to interact with this interface as it is provided to them in a CBM implementation. This can be accomplished simply in several ways, including inheritance from a base class in object oriented languages. The writer can then override a calling function that accepts an object providing method access to the outputs and any inputs. It is in this way that third-party code compiled into DLLs can be incorporated into the system transparently.

3. Interface Definitions
(continued)

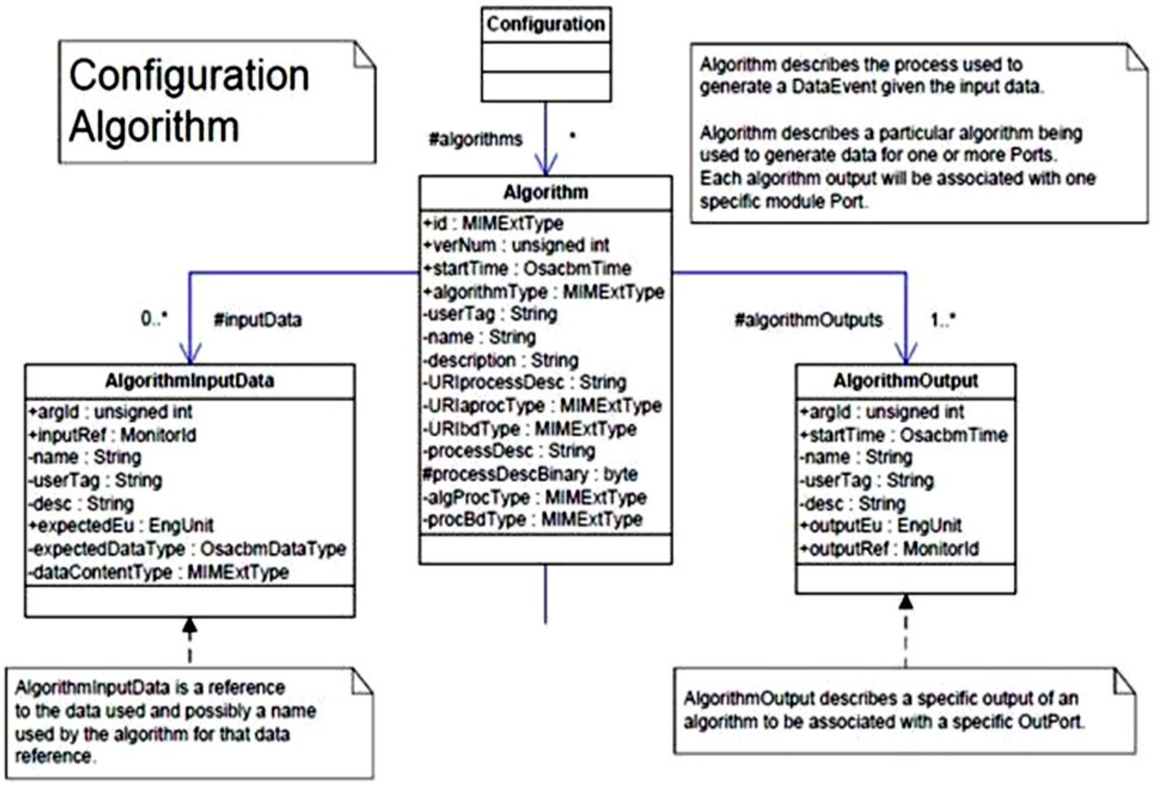
Configuration Algorithm
Configuration
#algorithms
*
Algorithm
+id : MIMExtType
+verNum : unsigned int
+startTime : OsacbmTime
+algorithmType : MIMExtType
-userTag : String
-name : String
-description : String
-URIprocessDesc : String
-URIaprocType : MIMExtType
-URIbdType : MIMExtType
-processDesc : String
#processDescBinary : byte
-algProcType : MIMExtType
-procBdType : MIMExtType
0..*
#inputData
AlgorithmInputData
+argId : unsigned int
+inputRef : MonitorId
-name : String
-userTag : String
-desc : String
+expectedEu : EngUnit
-expectedDataType : OsacbmDataType
-dataContentType : MIMExtType
#algorithmOutputs
1..*
AlgorithmOutput
+argId : unsigned int
+startTime : OsacbmTime
-name : String
-userTag : String
-desc : String
+outputEu : EngUnit
+outputRef : MonitorId
Algorithm describes the process used to generate a DataEvent given the input data.
Algorithm describes a particular algorithm being used to generate data for one or more Ports. Each algorithm output will be associated with one specific module Port.
AlgorithmInputData is a reference to the data used and possibly a name used by the algorithm for that data reference.
AlgorithmOutput describes a specific output of an algorithm to be associated with a specific OutPort.

Figure D: Algorithm Configuration in OSA-CBM UML Specification 3.3.0.

Summary

Implementation of the open architectures described herein is a recommended practice by AWEA. These architectures enable widespread and broad cooperation across the industry to enable improved capability and performance of wind turbine system.

References

[1] *Open System Architecture for Condition-Based Maintenance*, OSA-CBM 3.3.0, 2010.
[2] *Condition Monitoring and Diagnostics of Machines -- Data Processing, Communication and Presentation -- Part 2: Data Processing*, ISO-13374:2007, 2007